Enhanced Intrusion Detection in IIoT Networks: A Lightweight Approach with Autoencoder-Based Feature Learning

Tasnimul Hasan, Abrar Hossain, Mufakir Qamar Ansari and Talha Hussain Syed
Department of Electrical Engineering and Computer Science, The University of Toledo, Toledo, OH, U.S.A.
{*tasnimul.hasan, abrar.hossain, mufakir.ansari, talhahussain.syed*}*@utoledo.edu*

Keywords: Industrial Internet of Things, Intrusion Detection System, Autoencoder, Edge Computing, Lightweight Machine Learning.

Abstract: The rapid expansion of the Industrial Internet of Things (IIoT) has significantly advanced digital technologies and interconnected industrial sys- tems, creating substantial opportunities for growth. However, this growth has also heightened the risk of cyberattacks, necessitating robust security measures to protect IIoT networks. Intrusion Detection Systems (IDS) are essential for identifying and preventing abnormal network behaviors and malicious activities. Despite the potential of Machine Learning (ML)-based IDS solutions, existing models often face challenges with class imbalance and multiclass IIoT datasets, resulting in reduced detection accuracy. This research directly addresses these challenges by implementing six innovative approaches to enhance IDS perfor- mance, including leveraging an autoencoder for di- mensional reduction, which improves feature learning and overall detection accuracy. Our proposed Decision Tree model achieved an exceptional F1 score and accuracy of 99.94% on the Edge-IIoTset dataset. Furthermore, we prioritized lightweight model design, ensuring deployability on resource-constrained edge devices. Notably, we are the first to deploy our model on a Jetson Nano, achieving inference times of 0.185 ms for binary classification and 0.187 ms for multiclass classification. These results highlight the novelty and robustness of our approach, offering a practical and efficient solution to the challenges posed by imbalanced and multiclass IIoT datasets, thereby enhancing the detection and prevention of network intrusions.

1 INTRODUCTION

The Industrial Internet of Things (IIoT) integrates traditional industrial processes with advanced digital technologies, enabling the seamless interconnection of sensors, devices, and systems across various sectors(Lynn et al., 2020). This integration enhances operational efficiency, productivity, and innovation through real-time data collection, analysis, and decision-making. IIoT is applied in industries such as manufacturing, energy, healthcare, and transportation, where it supports predictive maintenance, asset tracking, remote monitoring, and smart automation(Xu et al., 2018). However, the extensive interconnectivity in IIoT systems makes them vulnerable to cyber threats, exacerbated by weak authentication mechanisms and irregular security updates(Alani, 2023). The vast number of connected devices and diverse communication protocols create multiple entry points for cybercriminals(Hassan et al., 2019), exposing IIoT networks to malware, data breaches, denial-of-service (DoS) attacks, and advanced persistent threats (APTs)(Yugha and Chithra, 2020). These threats can lead to significant operational disruptions, financial losses, and compromise the safety of critical infrastructure(Hassija et al., 2019).

Intrusion Detection Systems (IDSs) are essential for detecting and preventing unauthorized access and abnormal behavior in networks(Elrawy et al., 2018). IDSs monitor network traffic in real time, identifying potential threats using methods like signature-based, anomaly-based, and hybrid detection(Buczak and Guven, 2015; Leu et al., 2015; Mirsky et al., 2018). While these methods are effective, they face challenges in the complex and large-scale IIoT environments(Zarpelão et al., 2017), highlighting the need for more advanced IDS solutions. Moreover, current IDSs perform well with a limited number of classes and balanced data(Karatas et al., 2020). However, real-world scenarios often involve more classes and significant data imbalance, where infrequent yet critical classes may be overlooked. This imbalance can degrade detection performance, increasing the risk of

Hasan, T., Hossain, A., Ansari, M. Q. and Syed, T. H.
Enhanced Intrusion Detection in IIoT Networks: A Lightweight Approach with Autoencoder-Based Feature Learning.
DOI: 10.5220/0013203700003944

In *Proceedings of the 10th International Conference on Internet of Things, Big Data and Security (IoTBDS 2025)*, pages 207-214
ISBN: 978-989-758-750-4; ISSN: 2184-4976

undetected security threats. To address these challenges, performing machine learning (ML) inference on edge devices is crucial. Edge inference allows real-time traffic analysis, reducing latency and reliance on centralized systems while enhancing privacy and security. Lightweight ML models designed for edge environments operate efficiently under resource constraints, enabling robust intrusion detection even in remote or bandwidth-limited IIoT settings(Hossain et al., 2025).

To address these challenges, we make the following contributions in this paper:

- We propose a novel intrusion detection system (IDS) that leverages an autoencoder for dimensionality reduction, effectively addressing challenges with imbalanced and multiclass IIoT datasets while enhancing feature learning.
- We provide robust evidence of our IDS's effectiveness, achieving an exceptional F1 score and accuracy of 99.94
- We highlight the practical applicability of our approach by being the first to deploy the top-performing IDS models on a Jetson Nano, achieving fast inference times of 0.185 ms for binary classification and 0.187 ms for multiclass classification, making it suitable for real-world edge computing environments.

2 RELATED WORK

A notable approach in IoT/IIoT network security is the Lightweight Stacking Ensemble Learning (SEL) method combined with Feature Importance (FI) for dimensionality reduction. This method reduces storage requirements by 86.9% while maintaining a high classification accuracy of 99.96% (Abdulkareem et al., 2024). SEL's performance, particularly in multiclass classification scenarios, surpasses traditional models like Decision Trees and SVMs (Hassini et al., 2024). Similarly, a CNN1D model offers an end-to-end approach for intrusion detection, achieving 99.96% accuracy across 15 attack classes (Saadouni et al., 2023). Its strength lies in efficiently handling feature extraction and classification, making it valuable in real-time Industrial IoT environments Furthermore, a Self-Attention-based Deep Convolutional Neural Network (SA-DCNN) enhances feature prioritization, improving detection in IIoT networks with an accuracy of 99.95% on the Edge-IIoTset (Alshehri et al., 2024). Deploying edge ML and HPC

The Edge-IIoTset dataset stands out for its comprehensive coverage of IoT/IIoT devices and protocols, supporting both centralized and federated learning models. It is more versatile compared to MQTTset and WUSTL-IIoT-2021, which have narrower focuses (Ferrag et al., 2022). Hybrid models like CNN-LSTM and CNN-GRU are also effective in IIoT security. The CNN-LSTM model excels in feature extraction and sequence prediction, particularly for complex attacks like MITM , while the CNN-GRU model optimizes time-series data classification, further enhancing IIoT network security (Saadouni et al., 2023). Additionally, the Multi-Head Attention-based Gated Recurrent Unit (MAGRU) model addresses data imbalance and complex class structures in IIoT, achieving high accuracy and precision on datasets like Edge-IIoTset and MQTTset (Ullah et al., 2023).

3 METHODOLOGY

This section outlines our methodology for designing an effective IDS for IIoT. Using the Edge-IIoTset dataset, we preprocess data through feature selection, normalization, and encoding. To tackle class imbalance, we introduce a cost-sensitive autoencoder that prioritizes underrepresented attack types. Finally, we implement a lightweight architecture optimized for deployment on resource-constrained edge devices, balancing accuracy and efficiency.

3.1 Dataset

The Edge-IIoTset dataset, comprising 2.2 million records from a seven-layer, ten-device setup, includes 1.6 million normal traffic instances and over 600,000 across 14 attack types, highlighting significant imbalance. Similarly, the MQTTset, with 541,000 records from eight sensors, also skews heavily toward normal traffic. Edge-IIoTset, featuring 61 novel detection-enhancing features, realistically represents imbalanced IIoT traffic, making it vital for developing robust intrusion detection systems.

Table 1 shows the distribution of attack types in our dataset. Below is a brief summary of these attacks:

- **Normal (71.65%):** Legitimate network traffic without malicious activity.
- **DDoS UDP (6.31%)**, **DDoS ICMP (3.53%)**, **DDoS TCP (2.60%)**, **DDoS HTTP (2.55%):** DDoS attacks using UDP, ICMP, TCP, or HTTP packets to overwhelm network resources.
- **SQL Injection (2.64%):** Inserting malicious SQL code to manipulate databases.

Table 1: Dataset Distribution of Edge-IIoTset.

	Traffic Class	Records
Normal	Normal	1,380,858 (71.65%)
Attack	DDoS_UDP	121,567 (6.31%)
	DDoS_ICMP	67,939 (3.53%)
	SQL_injection	50,826 (2.64%)
	DDoS_TCP	50,062 (2.60%)
	Vulnerability	50,026 (2.60%)
	Password	49,933 (2.59%)
	DDoS_HTTP	49,203 (2.55%)
	Uploading	36,915 (1.92%)
	Backdoor	24,026 (1.25%)
	Port_Scanning	19,983 (1.04%)
	XSS	15,066 (0.78%)
	Ransomware	9,689 (0.50%)
	Fingerprinting	853 (0.04%)
	MITM	358 (0.02%)
Total		**1,927,304 (100%)**

- **Vulnerability Scanner (2.60%):** Probing systems for security weaknesses.
- **Password (2.59%):** Attempting to crack or guess passwords.
- **Uploading (1.92%):** Unauthorized file uploads to compromise security.
- **Backdoor (1.25%):** Bypassing authentication for unauthorized access.
- **Port Scanning (1.04%):** Identifying open ports and services as a precursor to attacks.
- **XSS (0.78%):** Injecting malicious scripts into webpages (Cross-Site Scripting).
- **Ransomware (0.50%):** Encrypting data and demanding payment for decryption.
- **Fingerprinting (0.04%):** Gathering device information for targeted attacks.
- **MITM (0.02%):** Intercepting and altering communication between two parties (Man-in-the-Middle).

Algorithm 1: Training a Deep Autoencoder on Edge-IIoTset Dataset.

Data : Training and validation data in DataFrames
Result: Trained autoencoder model
1 Load training and validation samples into DataFrames;
2 Initialize and compile the model with Adam optimizer and MSE loss;
3 Define ModelCheckpoint for saving the best model and EarlyStopping to prevent overfitting;
4 Fit the model on the training set and validate on the validation set;
5 Save the best model;

3.2 Data Preprocessing

3.2.1 Feature Selection

Irrelevant columns were removed as they provided no predictive value, while columns with constant values and highly correlated features(correlation > 0.6) were dropped to reduce redundancy and computational complexity.

3.2.2 Label Encoding

Categorical variables were converted to numerical format using label encoding to ensure compatibility with machine learning algorithms.

3.2.3 Normalization

Features were normalized to a [0, 1] range using Min-Max Scaling, preventing scale dominance and improving training efficiency for sensitive models like neural networks.

3.3 Addressing Class Imbalance with Cost-Sensitive Autoencoder

As shown in Fig. 1 we introduce a cost-sensitive autoencoder to address the imbalanced nature of the Edge-IIoTset dataset. An autoencoder is a type of artificial neural network designed to learn efficient data representations by encoding inputs into a compressed latent space and then reconstructing the output as closely as possible to the original input.

The proposed autoencoder sets itself apart from existing approaches, such as variational autoencoders (VAEs), by focusing on architectural simplicity and addressing data imbalance directly. While VAEs aim

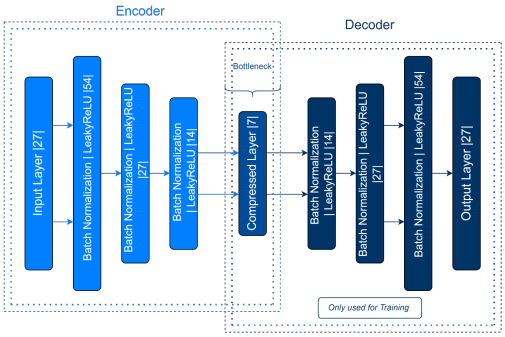


Figure 1: Architecture of the proposed cost-sensitive autoencoder.

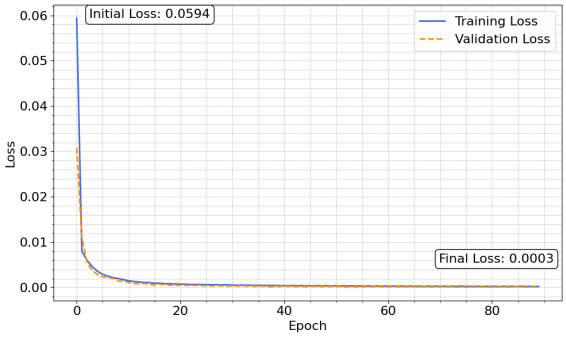


Figure 2: Autoencoder Loss Curve.

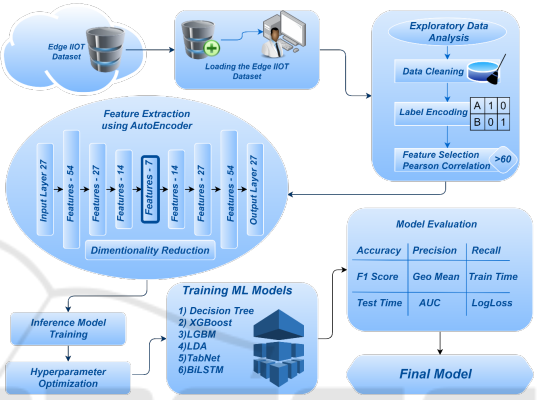


Figure 3: Proposed AutoEncoder based IDS Module.

to model data distributions through probabilistic latent representations, our approach is deterministic, prioritizing compact and discriminative feature learning for accurate reconstruction. By excluding the probabilistic layers used in VAEs, our autoencoder reduces computational complexity and is better suited for resource-constrained environments, such as edge devices. This deterministic approach also ensures stability in reconstruction, particularly when handling highly imbalanced datasets like Edge-IIoTset. Architecturally, our autoencoder is designed to efficiently handle the specific challenges posed by IIoT datasets. The encoder compresses the input feature space from 24 dimensions to a bottleneck layer of 6 dimensions, retaining critical features while discarding redundant information. This reduction is mathematically represented as:

$$h = f(x) = \sigma(Wx + b)$$

where W and b denote the weights and biases of the encoder, and σ is the nonlinear activation function. This compressed latent representation, h, captures the essential patterns of the input data. The decoder then reconstructs the input from h, expanding it back to the original dimensionality, as given by:

$$\hat{x} = g(h) = \sigma(W'h + b')$$

where W' and b' represent the weights and biases of the decoder. The reconstruction objective is to minimize the difference between the original input x and the reconstructed output $\hat{x}$. This difference is typically quantified using the mean squared error (MSE) loss function, defined as:

$$L = \frac{1}{N}\sum_{i=1}^{N} w_{y_i}(x_i - \hat{x}_i)^2$$

Here, N represents the number of inputs, x_i and $\hat{x}_i$ are the original and reconstructed values, and w_{y_i} denotes the class weight for the i-th instance.

To address the pronounced class imbalance in the Edge-IIoTset dataset, we employ a cost-sensitive learning mechanism using class weights. These weights are assigned inversely proportional to the frequency of each class, ensuring that minority classes, such as "MITM" and "Fingerprinting," receive greater attention during training. By amplifying the contribution of these underrepresented classes to the loss function, the autoencoder becomes more sensitive to detecting subtle variations in their patterns. This integration of class weighting into the training process eliminates the need for external methods like oversampling or data augmentation, which can introduce biases or increase computational costs. The results demonstrate the effectiveness of this strategy, as the model achieves superior performance on rare attack types without compromising overall accuracy. Compared to traditional autoencoders or VAEs, our method is more efficient, scalable, and specifically tailored for real-world deployment in IIoT environments. The combination of architectural optimization and cost-sensitive learning underscores the novelty of our approach and its ability to address the unique challenges of intrusion detection in imbalanced IIoT datasets.

Fig. 2 shows the training and validation loss curves of our autoencoder model which employs class weights inversely proportional to their frequencies,

emphasizing the learning of underrepresented attack types. As shown in Fig. 4 this approach enhances the model's ability to detect less frequent but critical attack types within the dataset, effectively addressing the class imbalance issue and improving the overall robustness of the intrusion detection system.

4 PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed approach.

4.1 Experimental Setup

The experiments were conducted on two platforms: a workstation with Ubuntu 20.04.3 LTS, 16 GB RAM, and a 2.20 GHz Intel Xeon processor, and a Jetson Nano with a 128-core Maxwell GPU and a 1.43 GHz Quad-core ARM A57 CPU. The workstation used Python 3.10.12, while the Jetson Nano used Python 3.7.

4.2 Evaluation Metrics

The performance evaluation of various models used is summarized in the following tables and the confusion matrix. We evaluated our models based on the following metrics.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1 = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (4)$$

4.3 Confusion Matrix

Table 2 presents the confusion matrix for several models, with the LGBM model achieving perfect detection accuracy (4820 TPs, 25620 TNs, 0 FP, 0 FN), while the XGB model also performs well (4818 TPs, 25619 TNs, 2 FPs, 1 FN). In contrast, the LDA model shows significantly weaker performance (1771 TPs, 24876 TNs, 3049 FPs, 744 FNs).

4.4 Results of Binary Classification

We present the results of binary classification in Table 3. LGBM achieves perfect performance across all metrics, while XGB and Decision Tree models also perform well but with slightly longer training times. In contrast, LDA shows significantly lower accuracy at 0.8754, indicating its inadequacy for this task. Although TabNet and Bi-LSTM deliver near-perfect scores, their high training times (1735.4461 seconds and 982.2894 seconds, respectively) suggest a trade-off between accuracy and computational efficiency.

Table 2: Confusion Matrix for Various Models (Absolute and Normalized Percentage Values).

Model	Pred. Normal (TP)	Pred. Attack (FP)	Pred. Normal (FN)	Pred. Attack (TN)
XGB	4818 (15.83%)	2 (0.01%)	1 (0.00%)	25619 (84.16%)
LGBM	4820 (15.83%)	0 (0.00%)	0 (0.00%)	25620 (84.17%)
LDA	1771 (5.82%)	3049 (10.02%)	744 (2.44%)	24876 (81.72%)
DT	4817 (15.82%)	3 (0.01%)	2 (0.01%)	25618 (84.16%)
TabNet	4816 (15.86%)	3 (0.01%)	0 (0.00%)	25545 (84.13%)
BiLSTM	4812 (15.81%)	8 (0.03%)	0 (0.00%)	25620 (84.17%)

4.5 Results of Multi Class Classification

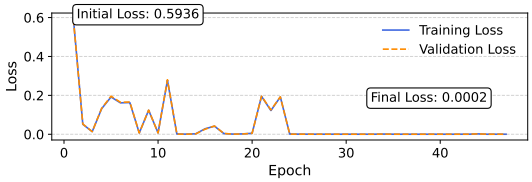
Table 4 summarizes our multi-class classification results using DT, XGB, LGBM, LDA, TabNet, and LSTM models, achieving 99.94% accuracy and F1 score, demonstrating strong anomaly detection capabilities. The Decision Tree model excels with near-perfect metrics and the fastest test time (0.0124 seconds), while LDA shows the lowest performance (0.4663 accuracy). XGB and LGBM achieve moderate accuracy but have longer training times. Although TabNet and LSTM perform well, their lengthy training times, particularly TabNet's 4436.6573 seconds, highlight the trade-offs.

4.6 Edge Inference Results

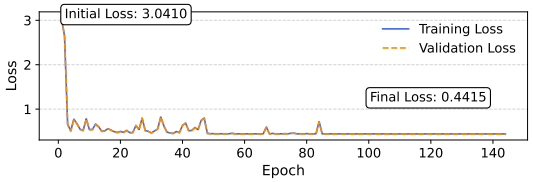
To demonstrate the lightweight nature of our model, we conducted inference tests using our best-performing models, Decision Tree and LGBM, on a Jetson Nano. For binary classification, the total inference time was 5.62 seconds, with an average time per instance of 0.184 ms. For multiclass classification, the total inference time was 5.70 seconds, with an average time per instance of 0.187 ms.

4.7 Comparison with Other Works

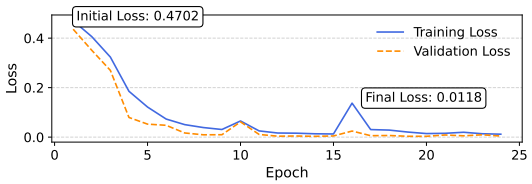
We compared the performance of our model against four recent intrusion detection techniques: (Alshehri et al., 2024), (Douiba et al., 2023), (Ferrag et al.,



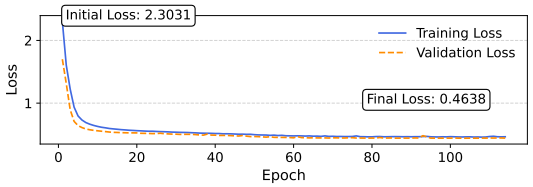
(a) Tabnet/Binary Loss curve



(b) Tabnet/Multi Loss curve



(c) BiLSTM/Binary Loss curve



(d) BiLSTM/Multi Loss curve

Figure 4: Training and Validation Loss for Tabnet and BiLSTM for Binary and Multiclass Classification.

Table 3: Summary of model performance (Binary).

Algorithm	Accuracy	Precision	Recall	F1	AUC	Geo Mean	Train Time (s)	Test Time (s)
LGBM	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0752	0.1099
XGB	0.9999	0.9999	1.0000	0.9999	0.9996	0.9998	1.0283	0.0597
DecisionTree	0.9998	0.9999	0.9999	0.9999	0.9994	0.9996	1.1718	0.0042
LDA	0.8754	0.8908	0.9710	0.9292	0.4487	0.5973	0.1600	0.0072
TabNet	0.9999	0.9999	1.0000	0.9999	1.0000	0.9997	1735.4461	2.7580
Bi-LSTM	0.9997	0.9997	1.0000	0.9998	0.9998	0.9920	0.9990	982.2894

Table 4: Summary of model performance (Multi Class).

Algorithm	Accuracy	Precision		Recall		F1 Score		Geo Mean	Train Time(s)	Test Time(s)
		Macro	Weighted	Macro	Weighted	Macro	Weighted			
DecisionTree	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9994	0.9997	3.7432	0.0124
XGB	0.8181	0.8106	0.8233	0.7974	0.8181	0.801	0.8183	0.8872	33.9725	0.8476
LGBM	0.7946	0.7915	0.8058	0.7737	0.7946	0.7768	0.795	0.8731	18.5499	4.8402
LDA	0.4663	0.3762	0.4359	0.3677	0.4663	0.3333	0.4261	0.5946	0.2051	0.0178
TabNet	0.7756	0.7751	0.7905	0.7488	0.7756	0.7548	0.7769	0.8584	4436.65	—
LSTM	0.7729	0.7682	0.7882	0.74	0.7729	0.7279	0.7567	0.8532	958.55	3.1581

2022), and (Ullah et al., 2023). While these methods exhibit promising results, they also face significant limitations, particularly in handling imbalanced datasets and multiclass classification scenarios, which are prevalent in IIoT environments. For instance, approaches such as (Alshehri et al., 2024) and (Douiba et al., 2023) struggle with class imbalance, often prioritizing majority classes while neglecting minority classes. This results in lower F1-scores and reduces their effectiveness in capturing nuanced patterns of intrusion. On the other hand, models like (Ferrag et al., 2022) and (Ullah et al., 2023) achieve high accuracy through complex architectures but fail to address the computational constraints of edge devices, making them unsuitable for real-time IIoT applications. In contrast, our approach not only outperforms these techniques but also addresses these limitations comprehensively. By leveraging an autoencoder for dimensionality reduction and feature learning, our model effectively mitigates the challenges posed by class imbalance. This is evident from our superior F1-score, as shown in Table 4, which demonstrates our ability to accurately classify both majority and minority classes. Additionally, while the accuracy

metric is commonly used, it can be misleading in imbalanced settings; our model achieves an outstanding accuracy of 99.94%, striking a balance between precision and robustness that outperforms existing methods. Furthermore, our approach is novel in its emphasis on practical deployment. Unlike prior work, our model introduces a lightweight architecture explicitly designed for edge environments, ensuring scalability and efficiency. We are the first to deploy an intrusion detection model on a Jetson Nano, achieving exceptionally low inference times of 0.185 ms for binary classification and 0.187 ms for multiclass classification.

Table 5: Comparison of Techniques and Performance Metrics.

Authors	Techniques	Accuracy	F1	Edge
Alshehri et al.	SA-DCNN	99.95%	99.53%	No
Douiba et al.	GB & DT	100%	99.50%	No
Ferrag et al.	DT, RF, SVM, KNN, DNN	94.67	99%	No
Ullah et al.	MAGRU	99.97%	99.64%	No
Our Approach	**DT, XGB, LGBM, LDA, TabNet, LSTM**	**99.94%**	**99.94%**	**Yes**

This capability demonstrates our model's suitability for real-world IIoT applications, where low-latency and resource efficiency are paramount. These contributions collectively establish our method as a robust, high-performance, and deployable solution that addresses the critical challenges faced by state-of-the-art intrusion detection systems.

5 CONCLUSION

Our Intrusion Detection System (IDS) achieved a 99.94% F1 score on the Edge-IIoT dataset, overcoming significant data imbalance. This success is attributed to our autoencoder-based methodology, which enhances feature learning and detection accuracy while being lightweight for edge inference, making it suitable for real-world IoT environments. Pretraining the autoencoder enabled efficient knowledge transfer, further boosting performance. Looking ahead, we plan to expand our research by testing the model on additional datasets and diverse IoT settings, integrating advanced machine learning algorithms, and refining feature extraction techniques to detect more sophisticated attacks. We also aim to improve the model's robustness and scalability for real-time deployment in large-scale IoT networks.

REFERENCES

Abdulkareem, S. A., Foh, C. H., Carrez, F., and Moessner, K. (2024). A lightweight sel for attack detection in iot/iiot networks. *Journal of Network and Computer Applications*, 230:103980.

Alani, M. M. (2023). An explainable efficient flow-based industrial iot intrusion detection system. *Computers and Electrical Engineering*, 108:108732.

Alshehri, M. S., Saidani, O., Alrayes, F. S., Abbasi, S. F., and Ahmad, J. (2024). A self-attention-based deep convolutional neural networks for iiot networks intrusion detection. *IEEE Access*.

Buczak, A. L. and Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications surveys & tutorials*, 18(2):1153–1176.

Douiba, M., Benkirane, S., Guezzaz, A., and Azrour, M. (2023). An improved anomaly detection model for iot security using decision tree and gradient boosting. *The Journal of Supercomputing*, 79(3):3392–3411.

Elrawy, M. F., Awad, A. I., and Hamed, H. F. (2018). Intrusion detection systems for iot-based smart environments: a survey. *Journal of Cloud Computing*, 7(1):1–20.

Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., and Janicke, H. (2022). Edge-iiotset: A new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning. *IEEE Access*, 10:40281–40306.

Hassan, W. H. et al. (2019). Current research on internet of things (iot) security: A survey. *Computer networks*, 148:283–294.

Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., and Sikdar, B. (2019). A survey on iot security: application areas, security threats, and solution architectures. *IEEE Access*, 7:82721–82743.

Hassini, K., Khalis, S., Habibi, O., Chemmakha, M., and Lazaar, M. (2024). An end-to-end learning approach for enhancing intrusion detection in industrial-internet of things. *Knowledge-Based Systems*, 294:111785.

Hossain, A., Badawy, A.-H. A., Islam, M. A., Patki, T., and Ahmed, K. (2025). Hpc application parameter autotuning on edge devices: A bandit learning approach. *arXiv preprint arXiv:2501.01057*.

Karatas, G., Demir, O., and Sahingoz, O. K. (2020). Increasing the performance of machine learning-based idss on an imbalanced and up-to-date dataset. *IEEE access*, 8:32150–32162.

Leu, F.-Y., Tsai, K.-L., Hsiao, Y.-T., and Yang, C.-T. (2015). An internal intrusion detection and protection system by using data mining and forensic techniques. *IEEE Systems Journal*, 11(2):427–438.

Lynn, T., Endo, P. T., Ribeiro, A. M. N., Barbosa, G. B., and Rosati, P. (2020). The internet of things: definitions, key concepts, and reference architectures. *The Cloud-To-Thing Continuum: Opportunities and Challenges in Cloud, Fog and Edge Computing*, pages 1–22.

Mirsky, Y., Doitshman, T., Elovici, Y., and Shabtai, A. (2018). Kitsune: an ensemble of autoencoders for

online network intrusion detection. *arXiv preprint arXiv:1802.09089*.

Saadouni, R., Khacha, A., Harbi, Y., Gherbi, C., Harous, S., and Aliouat, Z. (2023). Secure iiot networks with hybrid cnn-gru model using edge-iiotset. In *2023 15th International Conference on Innovations in Information Technology (IIT)*, pages 150–155. IEEE.

Ullah, S., Boulila, W., Koubaa, A., and Ahmad, J. (2023). Magru-ids: A multi-head attention-based gated recurrent unit for intrusion detection in iiot networks. *IEEE Access*.

Xu, L. D., Xu, E. L., and Li, L. (2018). Industry 4.0: state of the art and future trends. *International journal of production research*, 56(8):2941–2962.

Yugha, R. and Chithra, S. (2020). A survey on technologies and security protocols: Reference for future generation iot. *Journal of Network and Computer Applications*, 169:102763.

Zarpelão, B. B., Miani, R. S., Kawakani, C. T., and De Alvarenga, S. C. (2017). A survey of intrusion detection in internet of things. *Journal of Network and Computer Applications*, 84:25–37.